

Les Boucles en Python

for et while

Pourquoi utiliser des boucles ?

- **Réduire la répétition** : Exécuter un bloc de code plusieurs fois sans le réécrire.
 - **Automatiser des tâches** : Par exemple, parcourir une liste d'éléments ou effectuer des calculs répétitifs.
-

La boucle for

- **Utilisée pour** : Itérer sur une séquence (listes, tuples, chaînes, etc.).
 - **Syntaxe de base** : `python for élément in séquence: # bloc de code exécuté pour chaque élément`
 - **Exemple** : `python fruits = ["pomme", "banane", "cerise"] for fruit in fruits: print(fruit)`
-

La boucle while

- **Utilisée pour** : Répéter un bloc de code tant qu'une condition est vraie.
 - **Syntaxe de base** : `python while condition: # bloc de code exécuté tant que la condition est vraie`
 - **Exemple** : `python compteur = 0 while compteur < 5: print("Compteur :", compteur) compteur += 1`
-

La boucle for avec range()

- **range()** : Génère une séquence de nombres.
 - **Syntaxe** : `python for i in range(début, fin): # bloc de code exécuté pour chaque i`
 - **Exemple** : `python for i in range(1, 6): print("Numéro :", i)`
-

La boucle infinie avec while

- **Attention !** : Une boucle while sans condition de sortie peut tourner à l'infini.
- **Exemple** : `python while True: print("Cette boucle tourne indéfiniment...") break # Ajouté pour stopper la boucle`

Utiliser break et continue dans les boucles

- **break** : Sort de la boucle immédiatement.
 - **continue** : Saute à l'itération suivante de la boucle.
 - **Exemple** avec break : `python for i in range(10): if i == 5: break print(i)`
 - **Exemple** avec continue : `python for i in range(10): if i % 2 == 0: continue print(i)`
-

Boucles imbriquées

- **Utilisation** : Une boucle à l'intérieur d'une autre boucle.
 - **Exemple** : `python for i in range(1, 4): for j in range(1, 4): print(i, "*", j, "=", i * j)`
-

Utiliser les boucles pour manipuler des listes

- **Ajouter à une liste** : `python nombres = [] for i in range(5): nombres.append(i * 2) print(nombres)`
 - **Modifier des éléments d'une liste** : `python nombres = [1, 2, 3, 4, 5] for i in range(len(nombres)): nombres[i] = nombres[i] ** 2 print(nombres)`
-

Pourquoi utiliser les boucles ?

- **Éviter la redondance** : Réduire la répétition de code.
 - **Manipuler des données** : Parcourir, filtrer et modifier des collections de données facilement.
 - **Améliorer l'efficacité** : Automatiser les tâches répétitives.
-

Merci !

Des questions ?